Lab Report #4
Hanz Chua U26738740
Viraj Amarasinghe U14806439
Jian Gong U42910460

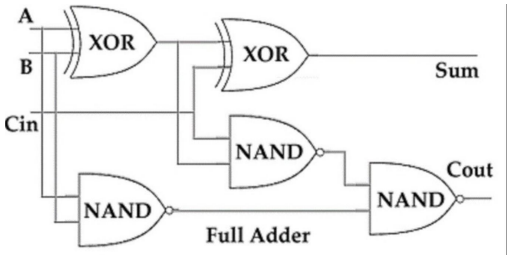
Introduction

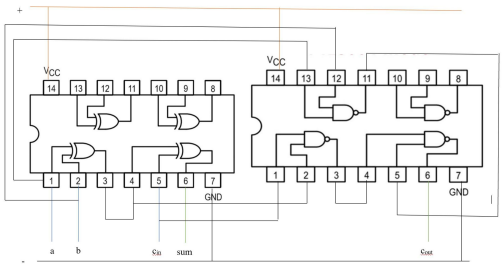
In this lab, we assembled a full adder circuit on a breadboard and implemented it as verilog code in EDA Playground. This lab's purpose was to introduce use to modeling basic combinational circuits using a hardware description language.

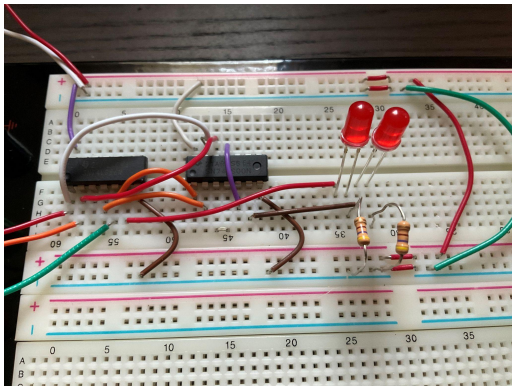
Methods and Materials

- Breadboard
- Wires
- 7400L500 Quad NAND Gate IC
- 74LS86 Quad XOR Gate IC
- 2 LED
- 2 470 ohm Resistor

1. First, we set up the verilog code on the EDA Playground. Go to the test bench window and copy the code at the bottom of this document and under the label Test Bench. Do the same for the module window with the code under the Module label. Make sure to select Synopsys VCS as the code's simulator. Make sure "Open EPWave after run" is checked off, so that we can observe the test case in timing diagram form. Then press run
2. We then must set up the circuit on a breadboard as follows.







Result

We observed that the output from the full adder in the EDA playground and the output from our breadboard full adder matched. The results came out as expected and contributed to the goal of the lab; to properly model basic circuits using hardware description language.

Conclusion

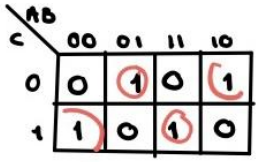
In conclusion, we tested the verilog code of a full adder, replicated it on a breadboard, and tested both of them to see if they yield the same result. Problems encountered included error in the test bench, as some code seemed unfamiliar. We also had issues in the breadboard full adder, since wires can be hard to keep track of

Questions

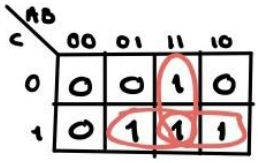
1.

A a	B b	C cin	s	cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

$S = \bar{A}\bar{B}C + \bar{A}B\bar{C} + A\bar{B}\bar{C} + ABC$

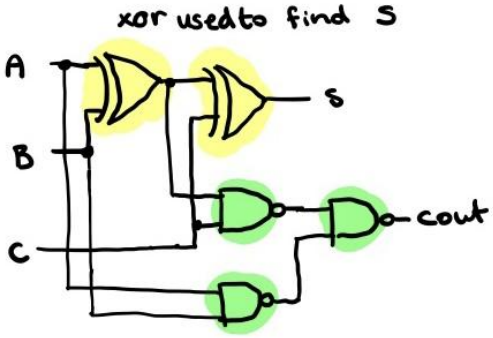


$\bar{B} + \bar{A}B\bar{C} + ABC$



$AB + BC + AC$

2.



nand uses $A \oplus B$ and nands to find cout

3. We can double negate the MSOP expression for Cout, which will yield ((AB)'(BC)'(AC)')'.

4.

Test Bench

```
module my_function_tb();
reg a, b, cin;
wire s, cout;
fulladder U0(a, b, cin, s, cout);
initial begin
$dumpfile("test.vcd");
$dumpvars;
$display("Starting simulation...\n");
$display("Time\ta\tb\tcin\ts\tcout\n");
$monitor("%2d\t%d\t%d\t%d\t%d\t%d",$time,a, b, cin, s, cout);
a = 0;
```

```
b = 0;
cin = 0;
#10 a = 0; b = 0; cin = 0;
#10 a = 0; b = 0; cin = 1;
#10 a = 0; b = 1; cin = 0;
#10 a = 0; b = 1; cin = 1;
#10 a = 1; b = 0; cin = 0;
#10 a = 1; b = 0; cin = 1;
#10 a = 1; b = 1; cin = 0;
#10 a = 1; b = 1; cin = 1;
#10 a = 0; b = 0; cin = 0;
#10 $finish;
end
endmodule
```

Module

```
module fulladder(a, b, cin, s, cout);
input a, b, cin;
output s, cout;
 wire out1,out2,out3;
 xor(out1, a, b);//out1 = A xor B, out2 = Cin and out1, out3 = A and
B
 and(out2, cin, out1);
 and(out3, a, b);

 xor(s, out1, cin);  //s = cin ^ (a and b)
 or(cout, out2, out3);     //cout = (cin and out1) + (a and b)

endmodule
```

5. a. Potential challenges with larger circuits include more input combinations. Errors are harder to trace in these larger circuits.

b.Propagation delay in the breadboard circuit was not accounted for in the simulation